

January 01, 2005

The effect of sudden server breakdown on the performance of a disassembly line

Surendra M. Gupta
Northeastern University

Gun Udomsawat
Northeastern University

Recommended Citation

Gupta, Surendra M. and Udomsawat, Gun, "The effect of sudden server breakdown on the performance of a disassembly line" (2005).
. Paper 114. <http://hdl.handle.net/2047/d10009887>

This work is available open access, hosted by Northeastern University.



Laboratory for Responsible Manufacturing

Bibliographic Information

Udomsawat, G. and Gupta, S. M., "The Effect of Sudden Server Breakdown on the Performance of a Disassembly Line", ***Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing V***, Boston, Massachusetts, pp. 93-102, October 23-24, 2005.

Copyright Information

Copyright 2005, Society of Photo-Optical Instrumentation Engineers.

This paper was published in Proceedings of SPIE (Volume 5997) and is made available as an electronic reprint with permission of SPIE. One print or electronic copy may be made for personal use only. Systematic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited.

Contact Information

Dr. Surendra M. Gupta, P.E.
Professor of Mechanical and Industrial Engineering and
Director of Laboratory for Responsible Manufacturing
334 SN, Department of MIE
Northeastern University
360 Huntington Avenue
Boston, MA 02115, U.S.A.

(617)-373-4846 **Phone**
(617)-373-2921 **Fax**
gupta@neu.edu **e-mail address**

<http://www.coe.neu.edu/~smgupta/> **Home Page**

The Effect of Sudden Server Breakdown on the Performance of a Disassembly Line

Gun Udomsawat and Surendra M. Gupta*

Laboratory for Responsible Manufacturing
334 SN, Department of MIE
Northeastern University
360 Huntington Avenue
Boston, MA 02115.

ABSTRACT

Product and material recovery relies on the disassembly process to separate target components or materials from the end-of-life (EOL) products. Disassembly line is especially effective when products in large quantity are disassembled. Unlike an assembly line, a disassembly line is more complex and is subjected to numerous uncertainties including stochastic and multi-level arrivals of component demands, stochastic arrival times for EOL products, and process interruption due to equipment failure. These factors seriously impair the control mechanism in the disassembly line. A common production control mechanism is the traditional push system (TPS). TPS responds to the aforementioned complications by carrying substantial amounts of inventories. An alternative control mechanism is a newly developed multi-kanban pull system (MKS) that relies on dynamic routing of kanbans, which tends to minimize the system's inventories while maintaining demand serviceability. In this paper we explore the impact of sudden breakdown of server on the performance of a disassembly line. We compare the overall performances of the TPS and MKS by considering two scenarios. We present the solution procedure and results for these cases.

Keywords: Breakdowns, Disassembly, JIT, Kanban.

1. INTRODUCTION

New methodologies play an important role in improving recovery rates of components and materials from end-of-life (EOL) products. Together with the advancement in reuse and recycling techniques, the recyclable materials and reusable components represent an important input in the production of both new and remanufactured products. In general, component recovery is more labor intensive and time consuming than material recovery because, in component recovery, additional care must be taken when separating components from EOL products to avoid their impairment. By allowing selective disassembly, valuable components from EOL products could be retrieved. Several studies have recently emerged that address various aspects of disassembly. Gungor and Gupta [4] provide a comprehensive survey of issues in environmentally conscious manufacturing and product recovery. Other studies on disassembly and product recovery have been presented by Brennan *et al.* [1], Gupta and McLean [9], Lambert [13] and McGovern and Gupta [15]. A recent book by Lambert and Gupta [14] is also helpful in understanding the areas of disassembly and disassembly modeling.

Disassembly line has recently become a subject of interest [2], [3], [15], [16] and [17]. A disassembly line consists of a series of disassembly workstations working in sequence. It is considered one of the best methods for the disassembly of EOL products in large quantity. However, a disassembly line faces numerous unique difficulties. Unlike an assembly line, which is a convergent process, a disassembly line is a divergent process [1]. It has significant inventory problems because of the disparity between the demands for certain parts or subassemblies and their yield from disassembly. Because of this disparity, a disassembly line must be controlled. Similar to an assembly line, there are two types of control mechanisms available in a disassembly line setting, viz., the push mechanism and the pull mechanism. A push mechanism is easy to implement but is not efficient in the disassembly environment. By its nature, it tends to generate large amounts of inventories. A pull mechanism, in theory, creates significantly fewer amounts of inventories. However, most production control tools that implement pull mechanisms in assembly line settings are not practical for the

*Correspondence: e-mail: gupta@neu.edu; URL: <http://www.coe.neu.edu/~smgupta/>
Phone: (617)-373-4846; Fax: (617)-373-2921

disassembly line settings. We demonstrate how the existing tools for the assembly line setting could be modified to implement them in a disassembly line setting.

Kanban is one of the most commonly used pull mechanism tools available. Hopp and Spearman [10] describe the kanban control mechanism in one-card and two-card environments. Gupta and Al-Turki [5], [6], [7] and Gupta *et al.* [8] propose the concept of the flexible kanban system (FKS) in various environments involving uncertainties and interruptions. They demonstrate that in such environments, FKS outperforms the traditional kanban system. Korugan and Gupta [12] suggest an adaptive way of implementing kanbans to a single-stage hybrid system. A hybrid system refers to a combination of two distinct lines, viz., a production line and a disassembly line.

In this paper we study the effect of sudden server breakdown on the performance of a disassembly line. We present a methodology to implement a kanban system in a disassembly line setting with the occurrence of a sudden breakdown. We provide a numerical example to illustrate the methodology and obtain results using simulation. We compare the performance of the pull system with that of the push system.

2. DISASSEMBLY LINE

Disassembly line consists of a series of workstations operating in a sequence to disassemble the end-of-life (EOL) products into subassemblies and/or components. In this section, we discuss the unique characteristics of a disassembly line. We also discuss the implications of a further complication that would occur due to the sudden breakdown of a server.

2.1 Typical Characteristics of a Disassembly Line

There are a couple of reasons for the disorderly fluctuation in the inventory levels in a disassembly line. The first is the arrival points and patterns of EOL products. Based on the type of EOL product, it may enter the disassembly line at any of the workstation, not just the first workstation. The second is the arrival points and patterns of demands. Depending on what is demanded, the demand could occur at any workstation, not just the last workstation. These two crucial differences make a disassembly line different from an assembly line. The inventory control mechanism must therefore be carefully addressed in order to manage these fluctuations. Figure 1 depicts a typical structure of a disassembly line.

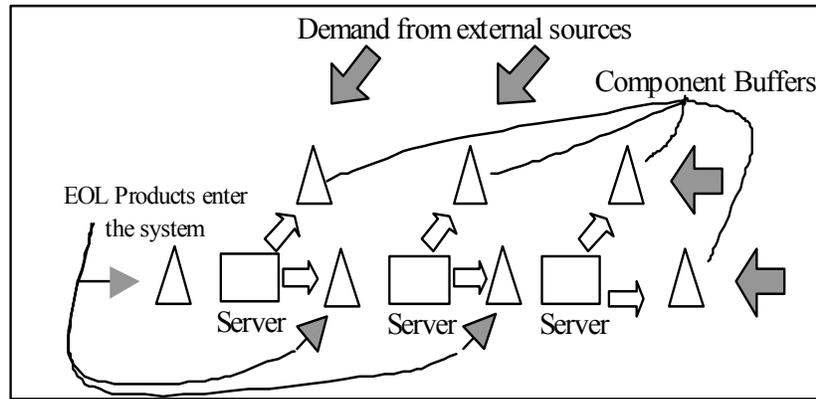


Figure 1. A Disassembly Line

In a disassembly line, the arriving products may consist of different combinations of components from a given set of components. From a set of N components, the total number of possible combinations of components, $Q_{(N)}$ is given by

$$Q_{(N)} = 2^N - N - 1 \quad (1)$$

For example, a set of 4 components (A, B, C, D) can produce up to 11 possible product combinations (viz., AB, ABC, ABCD, ABD, AC, ACD, AD, BC, BCD, BD, CD). By adding one more component to the set, the number of possible combinations increases to 26. It is therefore clear that the number of combinations increases exponentially with the increase in the number of components. The workstation where a product enters the disassembly line depends on the

combination of the components in the product. Thus for example, consider a disassembly line with three workstations. If component A is disassembled at workstation 1, component B is disassembled at workstation 2 and components C and D are disassembled at workstation 3, then a product arriving at the disassembly line consisting of components B, C, and D does not have to go to workstation 1 at all. It could enter the disassembly line directly at workstation 2. Considering the same example, if an arriving product consists of components A, C, and D, it would have to enter workstation 1. However, after getting processed at station 1, it could skip workstation 2 entirely. Both these situations destabilize the disassembly line by causing an overflow of materials at one workstation while starving some other workstations leading to undesirable fluctuations of inventory in the system. It is therefore crucial to balance the line and manage the materials flow of the line.

The arrival pattern of demand in a disassembly line is much more complicated than in a typical assembly line. The key reason is the multilevel arrival of demand. Demand can occur at any station of the disassembly line. In most assembly lines, demand arrives only at the last workstation. However, in that case, even if multilevel arrival of demand were considered, its effect would be benign because the product does not go forward from there on as it is taken off the line to fulfill the demand. In a disassembly line setting, however, the arrivals of external demand at workstations other than the last one creates a disparity between the number of demanded components and the number of partially disassembled products. Thus, if the system responds to every request for components, it would end up with a significant amount of extra inventory of components that are in low demand. All this creates chaos in the system. Since service level is important and it is necessary to maximize it, it becomes necessary to develop a good methodology to control the system and find a way to manage the extra inventory produced. In this paper, we use a modified version of a multi-kanban mechanism suitable for the disassembly situation to control the system.

2.2 Disassembly Line and Sudden Server Breakdown

Disassembly of EOL products in large quantity can be challenging when breakdowns exist. A sudden breakdown in a disassembly line leads to unusual blocking and starving of workstations. A pull system employing multi-kanban mechanism [17] offers a viable solution that is capable of keeping up with the demand rate in such situations without carrying large amounts of inventories.

3. THE TRADITIONAL PUSH SYSTEM

A push system refers to a line where the processing is scheduled according to predetermined arrival of materials. The schedule is based on the expected demand of finished items. In a disassembly environment, push system allows workstations to disassemble components as long as there are EOL products or subassemblies waiting in the input buffers. A server retrieves EOL products on a first come first serve basis. After a component is disassembled, it is kept in its corresponding component buffer where it waits for retrieval by demands. The residual subassembly is sent to the next workstation corresponding to the next component in the disassembly sequence. The push system provides higher levels of customer service because the system tends to build up large inventories. This seems to benefit the system when there are sudden breakdowns of servers. However, building up inventory is one of the least desirable ways to control a disassembly line.

4. THE MULTI-KANBAN SYSTEM

The processing in a pull system is triggered by the actual demand that causes a flow of materials throughout the system. The pull mechanism is designed to control the inventory levels. It relies on the consistency of raw materials supplies and agility of the server. It only produces when and where there is a need. This is the main reason why pull mechanism is more likely to perform better than push mechanism in a disassembly line. Among several tools to implement pull mechanism, kanban is one of the most commonly used. However, once implemented in a disassembly line setting, it is fraught with numerous uncertainties. A modification of the mechanism is therefore needed to improve its performance by reducing these difficulties and allowing the system to operate at its best.

4.1 Material Types

There are two basic types of materials in the system, viz., *components* and *subassemblies*. A component is a single item that cannot be further disassembled. It is placed in the component buffer waiting to be retrieved via a customer demand. On the other hand, a subassembly is something that can still to be disassembled. Subassembly is composed of at least two components. Both types of materials can be further distinguished as regular or overflow items. Regular items are

what customers or downstream workstations demand. In order to fulfill the demand, a server must disassemble the demanded component or subassembly. The residual item from this disassembly process that does not fulfill any request is called overflow item. Because the disassembly process is initiated by a single kanban, the overflow item will not have a kanban attached to it. However, the overflow item is routed in the same way as the regular item. The only difference between them is that the overflow item is given priority of being retrieved after it arrives at its buffer. It should be noted that, as long as there is an overflow item in the buffer, its demand would not initiate any further disassembly process. This will help the system eliminate any extra inventory first that is caused due to unbalanced demands

4.2 Kanban Types

Corresponding to material types, there are two basic types of kanbans in the system, viz., component kanbans and subassembly kanbans. A component kanban is attached to a disassembled component that is placed in the component buffer of the workstation where it is disassembled. Similarly, a disassembly kanban is attached to a residual subassembly that is placed in the subassembly buffer of the workstation where it was separated from the component. A component placed in a component buffer can be retrieved by an external demand. When authorized, a subassembly placed in the subassembly buffer is routed for disassembly to the next workstation based on its disassembly sequence.

At the first workstation, products arrive only from outside sources. However, at any other workstation i , where $1 < i \leq N-1$, there are two possible types of arrivals. The first type is a subassembly that arrives from an upstream workstation, called internal subassembly. There is always a subassembly kanban attached to an internal subassembly. The second type is a product (or subassembly) that arrives from outside sources, called external subassembly. There is no kanban attached to an external subassembly. This is also true of the products arriving from external sources to the first workstation. As long as there is an external product or subassembly available at an input buffer, the system will process it first before processing any available internal subassembly. This will avoid unnecessary pulling of an internal subassembly from an upstream workstation. Thus, the number of kanbans attached to internal subassemblies will remain constant throughout the process. Figure 2 illustrates the kanbans and materials flow in a disassembly line.

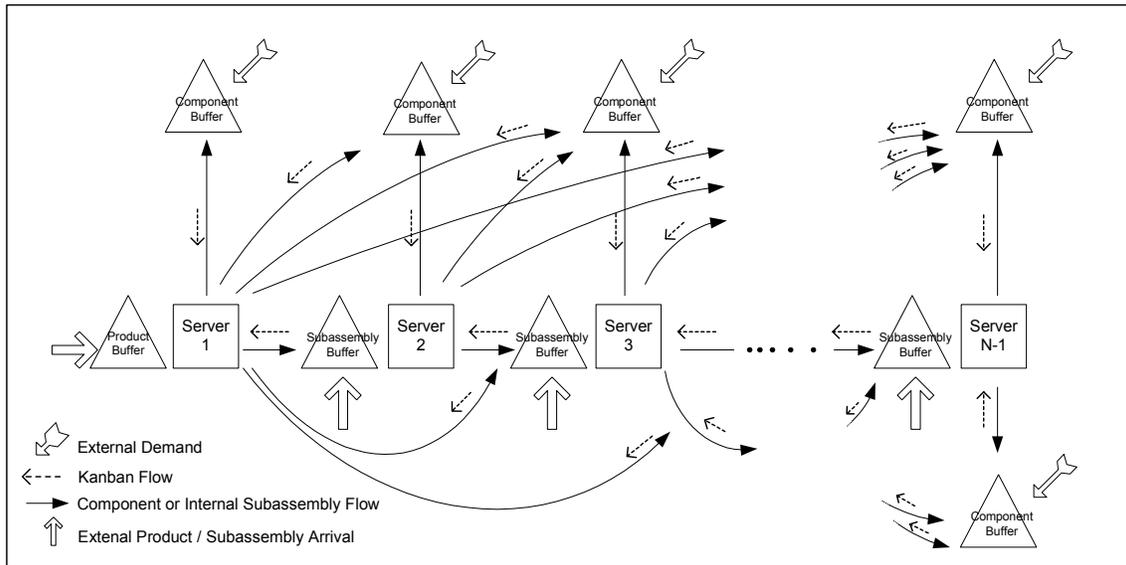


Figure 2. Kanbans and Materials Flows in a Disassembly Line

4.3 Kanban Routing Mechanism

Consider workstation j , where $1 \leq j \leq N-1$. When a demand for component j arrives at the component buffer of workstation j , one unit of component j is retrieved and the component kanban j attached to it is routed to the most desirable workstation. The procedure for determining the most desirable workstation to route component kanban j is given below. (Note that this procedure is not applicable to component kanbans $N-1$ and N . In both cases the kanbans are routed to the input buffer of the last workstation).

A component kanban originating from workstation j will be routed to a workstation i , where $1 \leq i < j$, or workstation j depending on the availability and the desirability of the subassembly that contains component j . Routing component kanban j to workstation i , where $1 \leq i \leq (j-1)$, will result in an immediate separation of component j from component i . Thus, the only subassembly located at the input buffer of workstation i that would be useful is a subassembly that contains only components i and j . If this type of subassembly exists in the input buffer of workstation i , then workstation i is qualified. Similarly, if there is at least one subassembly in the input buffer of workstation j , then workstation j is qualified.

Next, we need to select the most desirable workstation to route component kanban j to, among the qualified ones, such that, if chosen, will cause the least amount of extra inventory in the system. Choosing workstation i will increase the inventory level of component i by an additional unit. Thus, the best workstation i is the one that is most starving for its component. By checking the backorder level for demand i , we could determine the most starving workstation. If there is a tie, select the most downstream workstation. Choosing workstation j will create a residual subassembly that will be further disassembled at downstream workstations. If workstation j is chosen, then a proper subassembly must be chosen to disassemble. For example, if a backorder exists at the component buffer of workstation k , where $j < k \leq (N-1)$, then, if available, we might try to disassemble a subassembly that contains only components i and k . If more than one workstation qualify as starving workstations, then the one that is most starving among them is chosen. If there is a tie, then the most downstream workstation is selected.

We can now compare the starving levels of workstations i and j . If the highest starving level of workstation i is greater than or equal to the starving level of workstation j then we will route the component kanban j to workstation i , otherwise, we will route it to workstation j . Note that whenever an external subassembly is available, it will always be chosen first. Internal subassemblies will only be used when no external subassembly of the desired kind is available. Subassembly kanbans are routed in a fashion similar to component kanbans. Figure 3 shows a concept of the Multi-Kanban Mechanism.

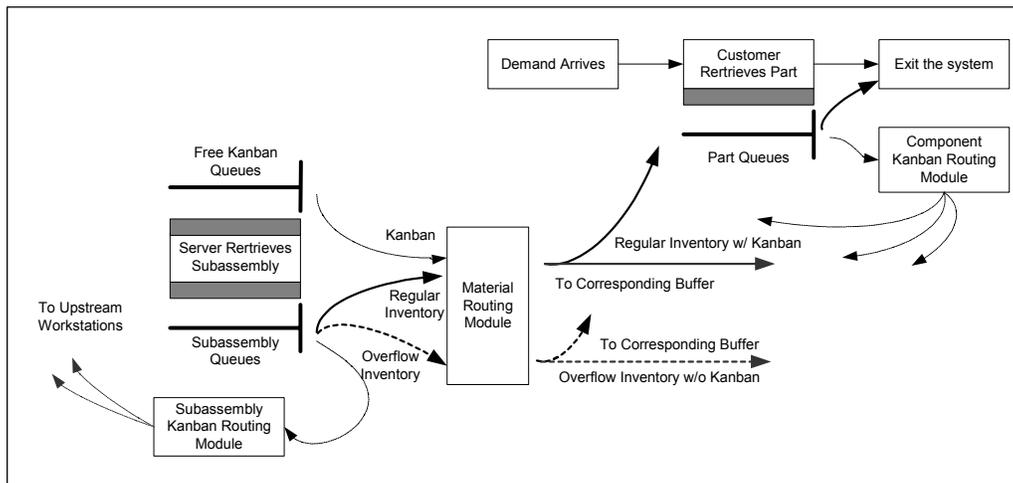


Figure 3. The Multi-Kanban Mechanism

4.4 Selection of Products

Because we allow multiple combinations of products, the worker may have several options when selecting the product for disassembly. If the authorization of disassembly is initiated by the subassembly kanban (j_x), which can occur only at workstation i , where $1 \leq i < j$, the workers will have no option but to select the subassembly that results in immediate separation of subassembly (j_x), viz., subassembly (ij_x). If the authorization of disassembly is initiated by component kanban j at workstation i , where $1 \leq i < j$, the worker will have to remove subassembly (ij) from the product buffer with no other options because the only subassembly that results in immediate separation of component j is the subassembly (ij). However, if the component kanban j arrives at workstation j , there are multiple options because every subassembly located in the product buffer contains component j and always results in immediate separation of component j . In this

case, we determine whether or not the residual that is created by the disassembly will result in overflow of inventory. We choose the subassembly (j_x) where x is the most desirable residual ranking based on the request of subassembly kanban x at workstation j (existing kanban x at the workstation j) or current inventory level of subassembly (component) x , respectively.

4.5 Determining the Kanban Level

The kanban level plays an important role in the multi-kanban mechanism as it maintains a proper flow of components and subassemblies at a desired level throughout the system. It can be determined by considering product arrival rate, demand arrival rate and disassembly time. The number of kanbans for both the *component kanban*, k_i and the *subassembly kanban*, k_j^* can be computed, at any point in the disassembly line, using the following general expressions:

$$k_i = \max(1, R_i / F_i) \quad (2)$$

$$k_j^* = \max(1, R_j^* / F_j^*) \quad (3)$$

where R_i is the *request rate* of component i , F_i is the *furnish rate* of component i , R_j^* is the *request rate* of subassembly j , and of F_j^* is the *furnish rate* subassembly j . These request rates and furnish rates can be calculated as follows:

$$R_i = d_i, \text{ for } 1 \leq i \leq N \quad (4)$$

$$F_i = \sum_{w=1}^i s_{(i,w)}, \text{ for } 1 \leq i \leq N \quad (5)$$

$$R_j^* = s_i, \text{ } i \text{ is the next component to be disassembled in the sequence} \quad (6)$$

$$F_j^* = a_j^* + \sum_{w=1}^{m-1} s_{(i,w)}, \text{ } i \text{ is the latest component disassembled in the sequence} \quad (7)$$

Where d_i is the demand arrival rate of component i , $s_{(i,w)}$ is the disassembly rate of component i at workstation w , s_j is the disassembly rate of subassembly j , a_j^* is the arrival rate of subassembly j (from external source), m is the current workstation index, N is the maximum number of component, and $N-1$ is the maximum number of workstation. For the case of *component kanban*, which is requested only from a single source, request rate is equal to the customer demand arrival rate. However, because the component kanban arrives from several sources in the system, the furnish rate is the summation of arrival rates from all possible sources. For the case of *subassembly kanban*, the furnish rate is influenced by both the disassembly rate and the external subassembly arrival rate. Thus, we take all external and internal arrival rates of subassemblies at the buffer into account. Similarly, the two requesting sources, viz., the demand for target component and the demand for residual subassembly affect the request rate. The number of kanbans is determined at the beginning of the disassembly process. It is clear that demand, supplies, disassembly time, and product structure, all affect the computation of the number of kanbans.

4.6 Coping with Sudden Breakdown of a Server

When there is a breakdown of a server, the mechanism copes with such situation by routing the kanban from the starving buffer to other candidate workstations. The comparison criteria remain unchanged. However, the breakdown workstation is excluded and does not qualify as a candidate. That is, the mechanism takes advantage of the availability of multiple candidates and requests the part from other workstations that are still operating. A workstation that is blocked by the breakdown is acceptable as a candidate. However, in a situation where the needed subassemblies are abundant at a workstation, it would most likely become the desirable workstation because the mechanism would consider it to be the one that has a tendency to generate the residual subassembly at a higher rate. Figure 4 demonstrates the example of a blocking workstation and starving buffers. Solid arrows show a sample of how a starving buffer C could receive component C from two candidate workstations (viz., 1 or 2) upstream from the blocking workstation 3. Also, in this case, candidates for component kanban B can be either workstation 1 or 2. The mechanism weighs the priority between the demand of residual parts at each workstation, viz. component A at workstation 1, and subassembly DEF or component B at workstation 2.

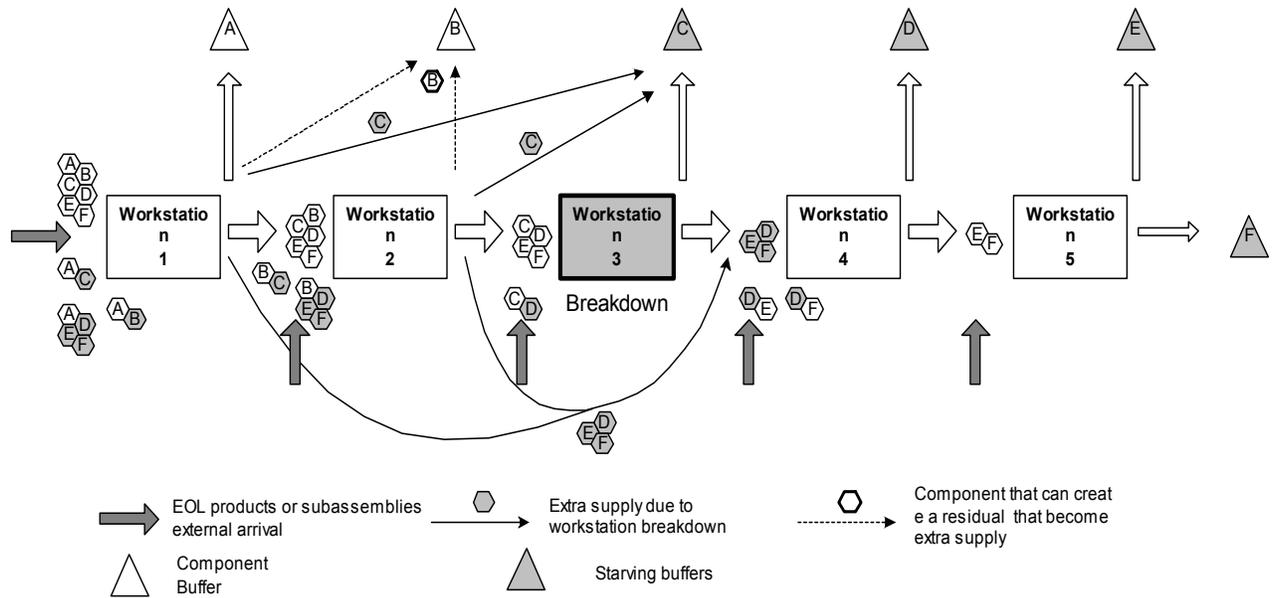


Figure 4. Coping with the Workstation Breakdown

5. CASE EXAMPLE AND SOLUTION METHODOLOGY

We consider a disassembly line with 4 workstations. There are six different EOL products (viz., ABCDE, ABC, AC, BCDE, BC, and CDE) made up of various combinations of components from a set of five possible components, viz., A, B, C, D, and E, that have to be disassembled to fulfill the demands for each of the components. The input location for an EOL product depends on its configuration. The input location is the most upstream workstation that disassembles the first component, according to the precedence relationships of that product. Only one type of component is disassembled at a given workstation except when there are only two components left in the product. To simplify the model, we assume that it takes the same amount of time to disassemble each component. At each workstation, there are two types of output buffers, viz., component buffer and subassembly buffer. The component disassembled at a workstation, s_i , is placed in the *component buffer*, B_i . The rest of the subassembly is routed to the *subassembly buffer*, B'_i corresponding to the next component to be disassembled. The subassembly buffer becomes the input buffer for the subsequent workstation to further disassemble the subassembly according to its disassembly sequence.

In studying the case example using simulation, the following assumptions were made:

- Customer backorder is allowed.
- External demand is for component only and can arrive at any workstation.
- Components must be disassembled according to their precedence relationships one type at a time until the last component in disassembly sequence is disassembled.
- Products and subassemblies may enter at any workstation along the line depending on its configuration.
- Only a single workstation breakdown is allowed.

We used ARENA® software [11] to simulate the model. We ran two sets of experiments representing the traditional push system and the multi-kanban system. For each experiment, we collected the data over a 24-hour period. In the push system, all arriving products are processed continuously in the order of their arrival. The demand is fulfilled as soon as the components are available. In the multi-kanban pull control system, we utilize smart-routing for subassembly kanban (as explained in the Kanban Routing Mechanism subsection) in order to reduce the inventory built up caused by disparity in demands among components. We also utilize product selection method (as explained in the Selection of Products subsection). In these experiments, statistics on the following two performance measures were collected: system's ability to fulfill demand and average inventory level (see Figures 5 and 6). Breakdowns are scheduled to occur at random. Time to repair is exponentially distributed. Both TPS and MKS systems experience breakdown and at the same location. We schedule up to 3 breakdowns during the 24 hours experimentation. Table 1 shows the input data for the disassembly line. The number of kanbans are calculated using the suggested method.

Table 1. Product Arrival Rates, Demand Arrival Rates, Disassembly Times, and Number of Kanban

Product, Subassembly, or Component	Mean EOL Product Arrival Rate (units/hour)	Mean Disassembly Time (minutes)	Mean Demand Arrival Rate (units/hour)	Number of Kanbans
ABCDE	4	-	-	-
ABC	4	-	-	-
AC	4	-	-	-
BCDE	1	-	-	1
CDE	1	-	-	1
BC	1	-	-	1
A	-	15	12	3
B	-	15	10	3
C	-	15	15	2
D	-	15	6	2
E	-	15	6	2

6. EXPERIMENTATION AND RESULTS

The system is experimented using two different scenarios, one with short, multiple breakdowns and the other with one long breakdown. For the first scenario, there are three short breakdowns that occur during the 24-hour period. Each breakdown requires an average of 15 minutes to repair. In the second scenario, a single breakdown occurs that needs 2 hours to repair.

The simulation results are shown in Figures 5 and 6. As is clear from the figures, in every scenario the multi-kanban mechanism significantly reduces average inventory while maintaining the components' demands. In TPS, the system builds up inventory in order to fulfill components' demands. A fluctuation in demands is coped well by the large amount of inventory. The multi-kanban mechanism deals with fluctuation among demands by routing the kanbans to the most suitable workstation. In the case example considered here, a component kanban C could be routed to either workstation 1 or workstation 2 or workstation 3. By examining the number of parts being requested in real time, the system selected the appropriate destination for the kanban. For the example considered, the system was able to reduce the inventory levels by an average of 78% while fulfilling components' demands using only three or kanbans in the system.

7. CONCLUSIONS

In this paper we explored the impact of sudden breakdown of server on the performance of a disassembly line. We compare the overall performances of the TPS and MKS by considering two scenarios. We presented the solution procedure and results for these scenarios. The MKS allows the system to meet the components' demands and stabilizes the fluctuations in the system's inventory levels even with the breakdowns.

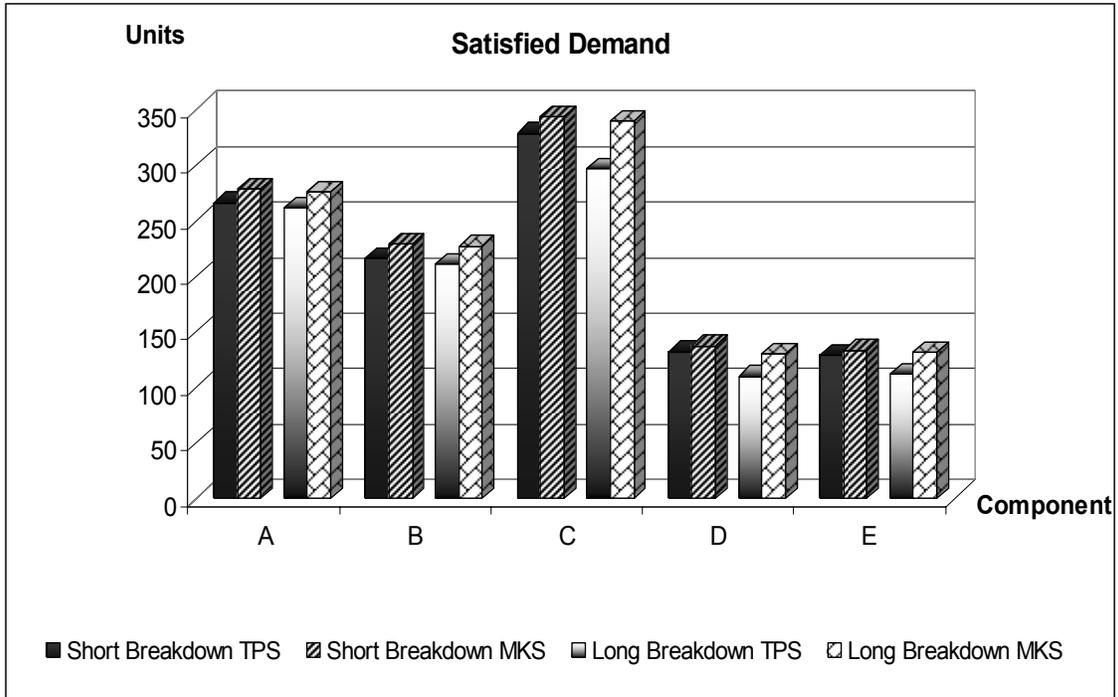


Figure 5. Number of Satisfied Demand for Multi-Kanban System and Tradition Push System

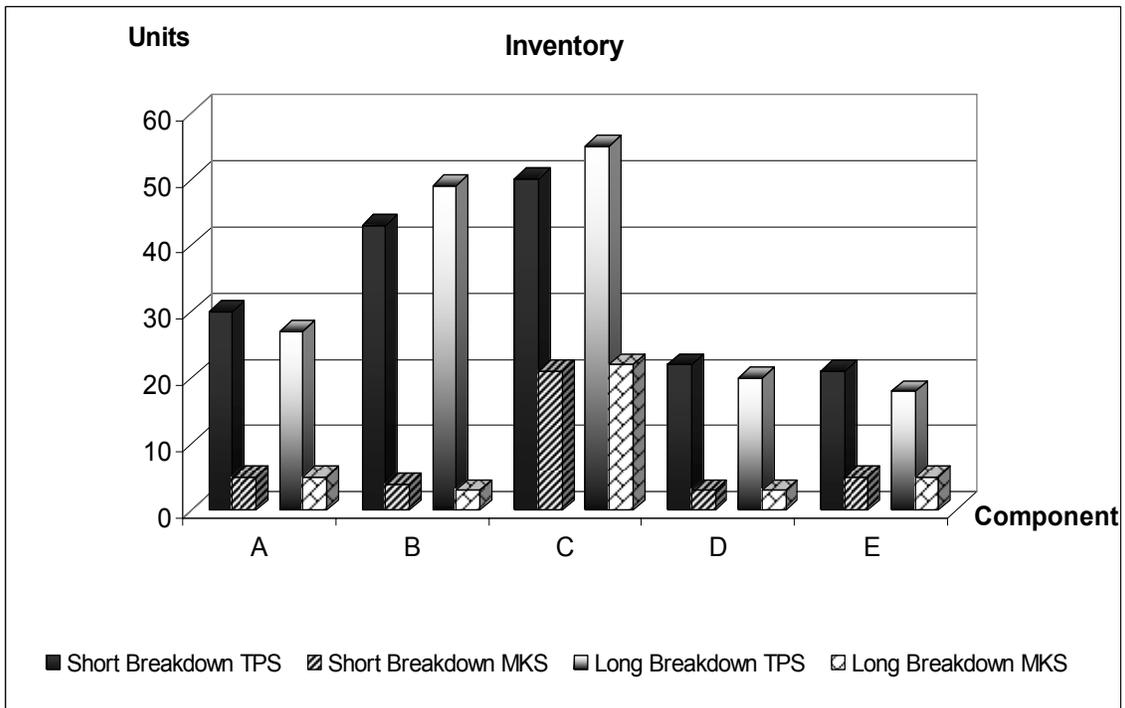


Figure 6. The Average Inventory Level

REFERENCES

- [1] Brennan L., Gupta S. M. and Taleb K. N., "Operations Planning Issues in an Assembly/Disassembly Environment", *International Journal of Operations Management and Production Management*, Vol. 14, No. 9, 57-67, 1994.
- [2] Gungor A. and Gupta S. M., "A Solution Approach to the Disassembly Line Balancing Problem in the Presence of Task Failures", *International Journal of Production Research*, Vol. 39, No. 7, 1427-1467, 2001.
- [3] Gungor A. and Gupta S. M., "Disassembly Line in Product Recovery", *International Journal of Production Research*, Vol. 40, No. 11, 2569-2589, 2002.
- [4] Gungor A. and Gupta S. M., "Issues in Environmentally Conscious Manufacturing and Product Recovery: A Survey", *Computer and Industrial Engineering*, Vol. 36, No. 4, 811-853, 1999.
- [5] Gupta S. M. and Al-Turki Y. A. Y., "Adapting Just-In-Time Manufacturing Systems to Preventive Maintenance Interrupts". *Production Planning and Control*, Vol. 9, No. 4, 349-359, 1998.
- [6] Gupta S. M. and Al-Turki Y. A. Y., "An Algorithm to Dynamically Adjust the Number of Kanbans in a Stochastic Processing Times and Variable Demand Environment", *Production Planning and Control*, Vol. 8, No. 2, 133-141, 1997.
- [7] Gupta S. M. and Al-Turki Y. A. Y., "The Effect of Sudden Material Handling System Breakdown on the Performance of a JIT System", *International Journal of Production Research*, Vol. 36, No. 7, 1935-1960, 1998.
- [8] Gupta S. M., Al-Turki Y. A. Y. and Perry R. F., "Flexible Kanban System", *International Journal of Operations and Production Management*, Vol. 19, No. 10, 1065-1093, 1999.
- [9] Gupta S. M. and McLean C. R., "Disassembly of Products", *Computers and Industrial Engineering*, Vol. 31, 225-228, 1996.
- [10] Hopp W. J. and Spearman M. L., "Factory Physics", Second Edition, McGraw-Hill, New York, 2001.
- [11] Kelton D. W., Sadowski R. P. and Sadowski, D. A., "Simulation with Arena®", WCB, McGraw-Hill, New York, 1998.
- [12] Korugan, A. and Gupta, S. M., "Adaptive Kanban Control Mechanism for a Single Stage Hybrid System", *Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing II*, Newton, Massachusetts, October 28-29, pp. 175-182, 2001.
- [13] Lambert A. J. D., "Disassembly Sequencing: A Survey", *International Journal of Production Research*, Vol. 41, No. 16, 3721-3759, 2003.
- [14] Lambert, A. J. D. and Gupta, S. M., *Disassembly Modeling for Assembly, Maintenance, Reuse, and Recycling*, Boca Raton, Florida, CRC Press, 2005.
- [15] M^cGovern S. M. and Gupta S. M., "Greedy Algorithm for Disassembly Line Scheduling", *Proceedings of the 2003 IEEE International Conference on Systems, Man, and Cybernetics*, Washington, DC, October 5-8, pp. 1737-1744, 2003.
- [16] Tang Y., Zhou M. and Caudill R., "A Systematic Approach to Disassembly Line Design", *Proceedings of the 2001 IEEE International Symposium on Electronics and the Environment*, Denver, Colorado, May 7-9, pp. 173-178, 2001.
- [17] Udomsawat, G., Gupta, S. M. and Al-Turki, Y.A.Y., "Multi-Kanban Model for Disassembly Line with Demand Fluctuation", *Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing III*, Vol. 5262, pp. 85-93, 2003.